LISTING OF CLAIMS

1. (Currently Amended)A computer readable medium encoded with a data structure, comprising:

a parameter definition for at least one expected input parameter; and

an instruction-based mechanism that is operative to identify information within an input source for each of the expected input parameters based on the respective definition for the expected input parameter and to process ~~populate~~ the expected input parameter based on ~~with~~ the information when the data structure becomes instantiated into an object, the input source comprising at least one live object.

2. (Original)The computer readable medium of claim 1, wherein the input source comprises a string.

3. (Original)The computer readable medium of claim 2, wherein the string comprises a part of a script.

4. (Original)The computer readable medium of claim 2, wherein the string comprises a part of a command string entered on a command line.

5. (Original)The computer readable medium of claim 1, wherein the parameter definition comprises a data type and a name for the expected input parameter.

6. (Original)The computer readable medium of claim 1, wherein the information comprises a value.

7. (Original)The computer readable medium of claim 6, wherein the parameter definition comprises a data type and a name for the expected input parameter, and wherein the mechanism further coerces the value into a converted value having the data type specified in the definition.

8. (Original)The computer readable medium of claim 1, wherein the input source comprises a set of objects.

9. (Original)The computer readable medium of claim 8, wherein the set of objects comprise .NET objects.

10. (Original)The computer readable medium of claim 1, wherein the input source comprises a precisely parseable stream.

11. (Original)The computer readable medium of claim 10, wherein the precisely parseable stream comprises an XML-based document.

12. (Original)The computer readable medium of claim 1, wherein the mechanism further identifies and populates each expected input parameter for each record within the input source.

13. (Original)The computer readable medium of claim 1, further comprising a mapping mechanism that is operative to associate a mapped name with the expected input parameter, wherein identifying the information is based on the mapped name.

14. (Original)The computer readable medium of claim 1, wherein the mechanism comprises a method inherited from a class provided within a runtime environment.

15. (Original)The computer readable medium of claim 1, wherein the parameter definition comprises a direct specification within the data structure.

16. (Original)The computer readable medium of claim 15, wherein the direct specification comprises a parameter declaration.

17. (Original)The computer readable medium of claim 1, wherein the parameter definition comprises an indirect specification associated with the data structure.

18. (Original)The computer readable medium of claim 16, wherein the indirect specification comprises a reference to an XML-based document that defines the at least one expected input parameter.

19. (Currently Amended)A computer-executable method for populating parameters declared within a data structure, the method comprising:

obtaining an expected name for a parameter, the expected name being assigned in a declaration for the parameter within a data structure;

identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;

retrieving a value associated with the label; and

assigning the value to the parameter.

20. (Original)The method of claim 19, wherein the expected name and the label are identical.

21. (Original)The method of claim 19, further comprising providing mapping information that defines an alias name for the expected name and identifying the label based on the alias name.

22. (Original)The method of claim 21, wherein the input source comprises a command string entered on a command line and the alias name is provided within the command string.

23. (Original)The method of claim 21, wherein the alias name is provided within a data store.

24. (Original)The method of claim 19, wherein the input source comprises an XML document.

25. (Original)The method of claim 19, wherein the input source comprises a database table.

26. (Original)The method of claim 19, wherein the input source comprises a command string entered on a command line.

27. (Original)The method of claim 19, wherein the input source comprises a script.

28. (Currently Amended)A system the handles input parameters, the system comprising:

a means for processing; and

a memory means, the memory means being allocated for a plurality of computer-executable instructions which are loaded into the memory means for execution by the means for processing, the computer-executable instructions performing a method comprising:

a means for obtaining an expected name for a parameter, the expected name being assigned in a declaration for the parameter within a data structure;

a means for identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;

a means for retrieving a value associated with the label; and

a means for assigning the value to the parameter.

29. (New) A computer-readable medium encoded with a data structure that provides a template for creating an application, the data structure comprising:

a name identifying an application;

at least one member;

a method;

a parent class from which the application derives, the parent class being provided by an object-based environment and providing processing that executes the method for each set of input received for the at least one member when the application is invoked, wherein the set of input comprises at least one live object.

30. (New) The data structure of claim 29, wherein the at least one member comprises an expected input parameter.

31. (New) The data structure of claim 30, wherein the parent class further provides validation processing on each set of input for the expected input parameter and does not execute the method for one set of input if the one set fails the validation processing.

32. (New) The data structure of claim 29, wherein the application comprises a command in a pipeline of commands and the set of input comprises results from a previous command in the pipeline of commands.

33. (New) The data structure of claim 29, wherein each set of input includes an identifier that associates the input with the member.

34. (New) The data structure of claim 29, wherein the parent class further provides a mapping process that allows a specified alias for the identifier.

35. (New) The data structure of claim 34, wherein the application comprises a command and the specified alias is provided as an argument to the command when the command is invoked

36. (New) The data structure of claim 35, wherein the command is invoked via an object-based command line environment.

37. (New) The method of claim 19, further comprising validating the value and wherein assigning the value to the parameter occurs if the value passes the validation.

38. (New) The computer readable medium of claim 1, wherein the live object is of a data type having a method, the method being directly invocable when processing the expected input parameter.